Mitigating Signaling Storms in 5G with Blockchain-assisted 5GAKA

Bohan Zhang
Cheriton School of Computer Science
University of Waterloo
Canada
bohan.zhang@uwaterloo.ca

Paul Zeinaty
Ecole Polytechnique,
Institut Polytechnique de Paris
France
paul.zeinaty@polytechnique.edu

Noura Limam Raouf Boutaba
Cheriton School of Computer Science
University of Waterloo
Canada
{noura.limam, rboutaba}@uwaterloo.ca

***Abstract*—This paper examines the registration signaling storm attack in 5G networks. This attack is initiated by massive registration requests and targets the 5G core network functions, leading to large-scale user service disruption. To mitigate this problem, we review the 5G Authentication and Key Agreement (5GAKA) protocol and highlight the impact of the exposure of the 5G core network (CN) to massive registration requests. We propose a blockchain-based authentication protocol to effectively block adversarial registration requests and secure the 5G network at a small cost. Our experiments reveal that our protocol is resistant to attacks and prove its superiority compared to a baseline mitigation approach.**

***Index Terms*—5G, security, 5GAKA, signaling storm, blockchain**

I. Introduction

Control signals are essential to manage network elements, allocate resources, and enable operations like handover, authentication, and billing. They precede any adaptation to changing conditions, performance optimization, and improvement to users' experience. Control traffic volume is typically measured in the number (and size) of transactions or messages exchanged between network elements. In 5G mobile networks, control signals are transmitted in the control plane (CP), while the data plane (DP) transmits actual user data. When the control plane signals' intensity exceeds capacity, it becomes a signaling storm. [1]

The signaling storm threat has been a common problem in 3GPP mobile broadband network technologies, dating back to the 3G mobile networks [2], [3]. It has attracted a lot of attention from the research community. Because control procedures between mobile devices and base stations typically involve massive volumes of radio signals, such as Radio Resource Control (RRC), Random Access Channel (RACH), and Paging, research on this particular challenge has mainly focused on the Radio Access Network (RAN) [4], [5]. However, experience shows that signaling storms are also a threat to the Core Network (CN). Telenor, the largest mobile network operator in Norway, reported an outage for calls and SMS due to the centralized mobile broadband server being overloaded by a signaling storm triggered by a simple reconfiguration of one voice server [6] in 2011. The outage impacted 3 million users for up to 18 hours, costing Telenor an 18 million USD loss. A straightforward way to prevent such outages is to increase the network capacity. For example, DoCoMo, Japan's largest telecom operator, spent 160 billion JPY in 2012 to increase the network capacity after an outage affecting 2.5 million users for 4 hours due to a signaling overhead [7]. However, in 2021, DoCoMo experienced another outage affecting 2 million users for 12 hours [8]. The outage occurred after several connections with IoT devices failed, and the devices were forced to re-register with the network. The simultaneous registrations flooded the CN, resulting in a delay in the authentication process. This caused the authentication procedure to time out and triggered subsequent re-attempts, gradually amplifying the signaling traffic and eventually overloading the network. It took DoCoMo hours to process the backlog. This outage exemplifies the danger of signaling storms and demonstrates that simply increasing network capacity to meet the rapid growth of edge devices is not an effective solution to signaling storm threats. This event also proved that re-registration had been a weak link in the cellular network system and could be exploited by malicious parties to threaten the security of today's 5G network. Gartner predicts that the number of connected IoT devices will reach 25 billion by 2030 [9]. The latest IoT Analytics report states that the number of global IoT connections grew by 18% in 2022 to 14.3 billion active IoT endpoints and is expected to grow by 16% in 2023, reaching 16.7 billion active endpoints [10]. The global cellular IoT market size was valued at USD 3.9 billion in 2021 and is projected to reach USD 15.4 billion by 2027, growing at a compound annual growth rate of 25.7% [11]. Cellular IoT devices can be weaponized to conduct signaling storm attacks in the context of 5G networks and incur significant damage, the same way their counterparts (e.g. IoT Mirai Botnet [12]) were used to launch various network attacks in the past. Indeed, this is a legitimate concern that must be addressed.

The contributions of this work are as follows:

- We formalize the registration signaling storm attack model. We show the importance of this problem through experimental evidence.
- We propose a blockchain-assisted registration protocol to mitigate this threat effectively. Our protocol can be integrated with both 5GAKA and EAP-AKA procedures and shall not affect authentication performance when

978-3-903176-59-1 ©2023 IFIP

there is no attack.
- We validate through rigorous performance evaluation the efficiency of our solution.

The remainder of the paper is organized as follows. We introduce the 5G architecture and 5GAKA procedure in Section II. Section III surveys and discusses related works. The registration signaling storm attack model is formalized and its potential damage is discussed in Section IV. Our proposed protocol and architecture are described and analyzed in Section V. Section VI provides the implementation details and evaluation results. Finally, Section VII concludes this paper.

II. Background

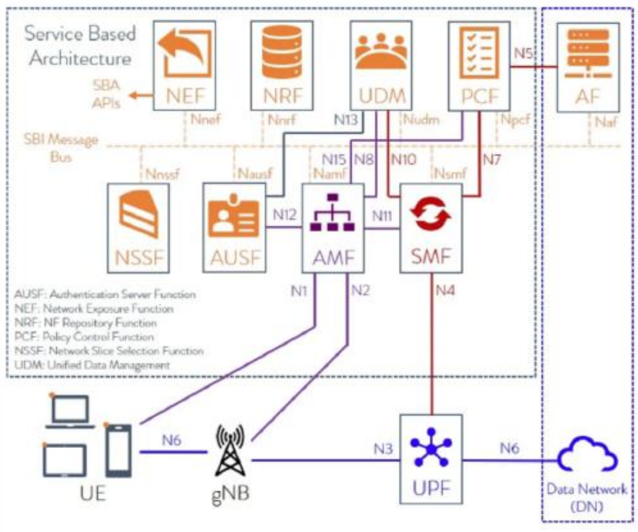
Service Based Architecture
SBA APIs
NEF
NRF
UDM
PCF
AF
N5
Nnef
Nnrf
N13
Nudm
Npcf
Naf
SBI Message Bus
N15
N8
N10
N7
Nnssf
Nausf
Namf
Nsmf
NSSF
AUSF
N12
AMF
N11
SMF
N1
N2
N4
AUSF: Authentication Server Function
NEF: Network Exposure Function
NRF: NF Repository Function
PCF: Policy Control Function
NSSF: Network Slice Selection Function
UDM: Unified Data Management
N6
UE
gNB
N3
UPF
N6
Data Network (DN)

Fig. 1: 5G service-based architecture [13]

One particular aspect of 5G is the 3GPP-defined Service-Based Architecture (SBA) in which the CP functionalities of the 5G network are delivered through a set of interconnected Network Functions (NFs), deployed using software from various sources and suppliers, with each NF authorized to access services of other NFs. Compared with prior 3GPP technologies, the SBA brings to 5G CN greater modularity, flexibility, and scalability.

Figure 1 shows the CP and DP separation in the 5G architecture, where DP is responsible for handling the delivery of user data, while CP manages network signaling, authentication, and resource allocation. As such, the resources dedicated to the CP are expected to be more constrained than those allocated to DP [1].

In the following, we will focus on the CP NFs and other components involved in the authentication procedure and describe the protocol.

A. SUPI and USIM

User Equipment (UE) is composed of two essential components: the Mobile Equipment (ME) and the Universal Integrated Circuit Card (UICC), which contains the Universal Subscriber Identity Module (USIM). The USIM is intended to securely store the subscriber's security-related context, including a globally unique Subscription Permanent Identifier (SUPI) and a long-term key K, and handle the computation incurred by the authentication protocol [14]. Once in possession of one's SUPI and the corresponding long-term key K, any party can authenticate itself to the 5G network on behalf of the legitimate UE. According to the 3GPP TS-33.501, the length and format of the SUPI are arbitrary. However, for legacy reasons, it is commonly assumed that the SUPI is equivalent, in length and format, to 4G International Mobile Subscriber Identity (IMSI). The IMSI is usually implemented as a 15-digit number. The first three digits are the Mobile Country Code (MCC), and the next three are the mobile network code (MNC). They, together, indicate a specific mobile operator in the world. The remaining digits are the Mobile Identification Number (MSIN), which uniquely identifies a subscriber to the operator's network. In 4G, when the UE registers to the network, the IMSI is sent in plaintext over the air. As such, the IMSI can be easily captured with the infamous IMSI catcher [15]. After one's IMSI (or SUPI) is leaked, the user's privacy is jeopardized because the attacker can exploit the Signaling System 7 (SS7) protocol to intercept calls and track the user's location [16]. To avoid this, the SUPI in 5G is first encrypted into the Subscriber Concealed Identifier (SUCI) and then sent over the air. To be more specific, a UE's MCC and MNC are sent in plaintext, but its MSIN is encrypted.

B. 5G Authentication Protocol

The 3GPP defined two standard protocols for UE registration, i.e., the 5GAKA and the Extensible Authentication Protocol Authentication and Key Agreement (EAP-AKA), with minor differences. In this section, we describe the workflow of the 5GAKA protocol following the 3GPP TS-33.501.

The NFs involved in the 5GAKA protocol are the Security Anchor Function(SEAF), the Authentication Server Function(AUSF), and the Unified Data Management Function(UDM). Because the SEAF is hosted in the Access and Mobility Management Function (AMF), AMF and SEAF are considered interchangeable in this work.

1) 5GAKA Initialization Phase: The initialization phase workflow is shown in Figure 2. The UE performs a SUPI

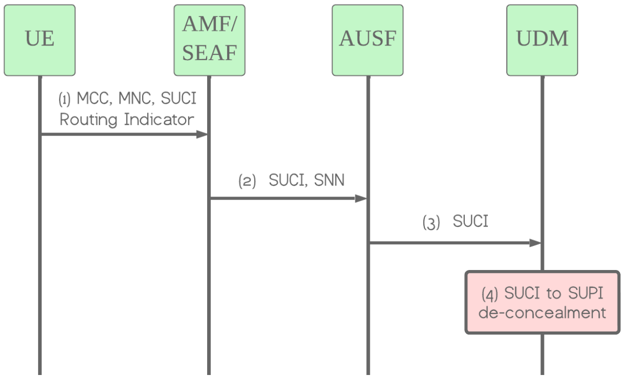
UE
AMF/ SEAF
AUSF
UDM
(1) MCC, MNC, SUCI Routing Indicator
(2) SUCI, SNN
(3) SUCI
(4) SUCI to SUPI de-concealment

Fig. 2: 5G-AKA Initialization

Algorithm 1 Elliptic Curve Integrated Encryption Scheme

```
Input: SUPI, $PK_{HN}$, $KDF$
Randomly generate key pair $SK_{UE}, PK_{UE}$.
$TK = PK_{HN} \cdot SK_{UE}$
$EK||ICB||MK = KDF(TK)$
$Cipher = Encrypt_{AES}(SUPI, EK)$
$MAC_{UE} = HMAC(Cipher, MK)$
$SUCI = PK_{UE}||Cipher||MAC_{UE}$
Return $SUCI$
```

to SUCI concealment following the Elliptic Curve Integrated Encryption Scheme (ECIES) showed in Algorithm 1 using the home network (HN)'s public key PK_{HN} and a local key generator function KDF. The UE sends to the SEAF the SUCI, MCC, MNC, and a routing indicator indicating which UDM should serve the UE. The SEAF then forwards the SUCI and the serving network name (SNN) to the AUSF. The AUSF temporarily stores the SNN and forwards the request to the UDM. Because the UDM holds the HN's secret key SK_{HN}, it performs a SUCI to SUPI de-concealment following ECIES decryption. After that, the UDM checks if the SUPI is a valid subscriber number of the HN.

2) 5GAKA Authentication Phase: The authentication phase follows a challenge-response authentication scheme shown in Figure 3. The UDM first prepares a random challenge $RAND$. Next, the UDM takes the input value $RAND$, K, and a synchronized sequence number; and feeds them to the key derivation function, resulting in $AUTN$ and session keys. The UDM sends SUPI along with a Home Environment Vector containing $RAND$, $AUTN$, and the expected response $XRES*$ to the AUSF. The AUSF then sends the SEAF a Serving Environment Vector (SEV) containing $RAND$, $AUTN$, and $HXRES*$, which is the hash of $XRES*$. Upon receiving $RAND$, $AUTN$ from the SEAF, the UE computes the response $RES*$ and sends it back to the CN. The AUSF sends SEAF the UE's SUPI and a session key K_{SEAF} after AUSF and SEAF verify the correctness of $RES*$, indicating successful authentication.

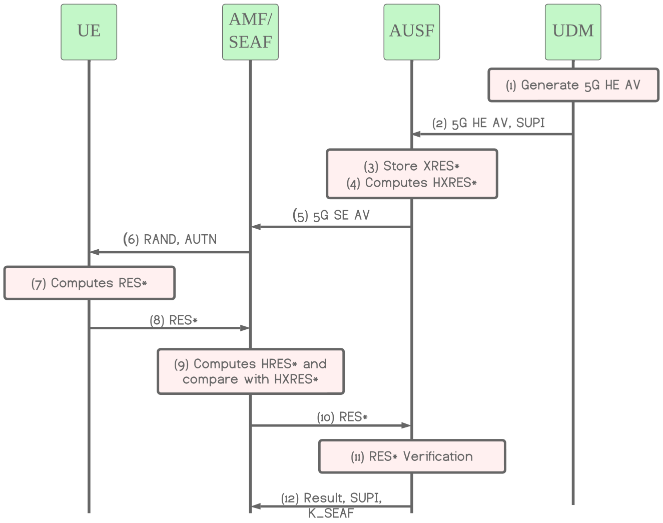

Fig. 3: 5G-AKA Authentication

III. Related Work

Several vulnerabilities and exploits against the 5G-AKA protocol were reported in the literature [17], [18]. These include the SUCI replay attack, linkability attack, and the lack of perfect forward secrecy vulnerability.

Although the aforementioned works have addressed different vulnerabilities of 5GAKA, the persistent registration signaling storm attack has yet to be thoroughly studied. Indeed, the more *centralized* nature of the UDM [19] makes 5GAKA vulnerable to such an exploit. Due to the distributed nature of blockchains, blockchain-based authentication protocols were suggested in the literature. [20] proposes to simply replace the UDM with a blockchain serving as a distributed database. In practice, such a method is hard to deploy because the UDM needs to manage many subscriber-related data, including billing, user location, and security-related UE context. Hence, a blockchain-based UDM brings synchronization and storage challenges to the operator. [21] also addresses the registration storm issue by solving the SUCI replay attack in a non-roaming scenario using a private blockchain. The main idea is that the UE will send the SUCI and a hash commitment of the next SUCI in the initialization phase. The commitment will be saved on the blockchain so that SUCI cannot be replayed. Although their results show the method is robust to signaling storm attacks, their approach is unrealistic to be considered. They completely removed the authentication phase in 5GAKA along with the long-term key K, and solely relied on gNB querying blockchain to perform the mutual authentication, resulting in the UE can hardly trust the HN. Unlike our solution, their change in 5GAKA will cause changes in other protocols, such as handover and PDU session establishment, because AMF and UDM no longer save the UE security context. Moreover, any compromised gNB will leak the HN secret key and can permanently deny the UE from the network by polluting the blockchain. Most blockchain-based existing solutions to 5GAKA vulnerabilities require writing at least one record on the blockchain per registration. This will undoubtedly introduce a significant storage overhead, and blocked synchronization may delay the authentication procedure.

In [22] signaling storm attacks are addressed at the RAN, without using blockchains, but rather using the O-RAN xApp framework. The proposed solution uses the timing advance (TA) parameter during the UE random access phase to identify a unique IoT device assuming the device is not moving. The xApp is responsible for making statistical analyses for every incoming random access request based on the history data. Such an idea can be compelling, considering the signal is stopped at the very beginning of the registration. However, the usage is very limited as it requires both gNB and UE to stay static, and UE must connect to the same gNB. Even under such an assumption, gNB does not have a promising role in identifying the malicious UE as the attack to the 5G core is distributed, and gNB can only have a partial view of the attack. Moreover, identifying a device using TA is inaccurate because the malicious device can easily change this value by tampering with the round trip time measurement.

Instead of detecting malicious users at the gNB, where the gNB can hardly have a global view, [23] proposes a machine learning detection system in the 5G core against 5G signaling storm attacks, including registration attacks. The 5G core is responsible for classifying malicious SUPIs using the history packets data feature, such as the time interval between two packets and the total count of packets during a given time. Although the result shows that the detection is ideal, such a method requires storing the history feature data of all UEs, which will introduce a huge storage burden. Moreover, they did not discuss how to mitigate the attack after classifying SUPI.

IV. Problem statement

In the following, we consider the 3GPP 5GAKA authentication procedure, even though our attack model and methodology apply both to 5GAKA and EAP-AKA. We aim to propose a 3GPP 5GAKA-compliant, efficient, and realistically deployable solution by 5G operators. We also assume the following:

1) SN and HN can securely communicate.
2) SN and HN securely store their cryptographic key pairs.
3) UE securely stores SUPI and long-term key K in USIM, unless compromised.
4) The air interface between UE and gNodeB is not secure. An attacker can monitor and tamper with the information transmitted on the N1 interface, the same as an adversary in the formal Dolev-Yao (DY) model [24].
5) As all cryptographic primitives, in the UE, are computed in the hardware USIM, using more cryptographic primitives would make the USIM more resourceful and more expensive. Thus, we assume the UE cannot do asymmetric decryption. [25]
6) The UE may or may not be in a roaming scenario. The UE will first communicate with SN's SEAF function in a roaming scenario. The UE will first communicate with HN's AMF function in a non-roaming scenario.
7) Most mobile operators implement UDM as a relatively centralized cloud-native function as it requires saving a huge amount of data. AMFs, on the other hand, are normally assumed to be relatively distributed as they can be easily deployed at the edge of the 5G CN.

A. Attack Model

With the increasing number of 5G cellular devices, it is realistically feasible for an adversary to compromise an important number of UEs (*e.g.*, cellular IoT devices) and use them to target an operator's UDM with a simple and powerful attack leveraging 5GAKA, and reproduce the damage induced by the Telenor [6] or DoCoMo incidents [7] [8], only to cite a few.

We may think of different attack scenarios leveraging 5GAKA to generate bogus registrations storm using compromised UEs. It is important to note that the compromised UEs can be geographically distributed. A registration signaling storm attack against an operator's network can be launched from another operator's network (a.k.a. roaming network). The attacker can simply specify the MCC, MNC, and routing indicator of the target network, and the roaming network will direct the registration message to the right place. Also, the attacker may use valid or invalid MSINs. For convenience, an MSIN is said to be valid (respectively invalid) if it exists (respectively does not exist) in the operator's UDM database. We distinguish three possible bogus registration scenarios:

1) The UE uses a valid MSIN, and then the UE can legally perform a full 5GAKA procedure.
2) The UE is replaying a previous SUCI, then the UE can make the UDM perform ECIES decryption and authentication vector generation and let the UDM hold the communication session.
3) The UE uses an invalid MSIN and carefully chooses the MCC and MNC, then the UE can make the UDM perform ECIES decryption.

Bogus registrations with valid MSIN (i.e., scenario 1) are relatively more harmful as they result in a higher overhead on the UDM and the CN. In fact, not only do they make the UDM perform a full 5GAKA procedure, but also they trigger several message exchanges between different functions in the 5G CN. However, such a method is relatively slower than the other two bogus registration scenarios where the UE does not need to interact with the HN. For the other two scenarios, the attacker can send massive requests within a short time frame and force the UDM to compute the ECIES, which is required to verify the subscriber's identity, hence consuming its processing power [26]. Such carefully crafted attacks can be triggered from multiple SNs, and the bogus registration signals will be ultimately forwarded to the targeted UDM. The attacker can also select a random subset of the compromised devices to register periodically or at random times, making them harder to detect.

B. Attack Outcome

When the 5G CN is under a registration storm attack, it is expected to experience a spike in traffic due to the increase in the number of registration requests. Besides, the AMF, SMF, and AUSF experience a temporary memory spike as they need to maintain temporary session management data. As these NFs are virtualized, their computation power and storage may not be designed to handle unexpected traffic, leading to the risk of crashing. More importantly, the SUPI can only be revealed at the more ”centralized” UDM; the massive number of incoming requests will require significant cryptographic computation, draining the UDM's computation power away from the legitimate users' requests. As the 5GAKA protocol happens entirely in the CP, which has more constrained resources than the DP, the massive amount of incoming traffic also causes traffic congestion, delaying legitimate users' requests and leading to a denial of service for legitimate users.

Overall, a registration storm attack is a distributed denial of service (DDoS) attack whose volume and impact can be amplified in two ways:

- Legitimately: by legitimate UEs whose registrations are delayed and repeatedly re-attempt registration, after a timeout.
- Adversely: once the 5GAKA procedure is completed (registration scenario 1), the adversary can trigger PDU session establishment and then trigger the de-registration procedure, causing more signals inside the CP. Each PDU session establishment involves the RAN and 5 NFs to exchange 17 messages in the CP. Moreover, one UE is allowed to trigger up to 15 PDU sessions. Finally, the de-registration procedure involves the RAN and 5 NFs exchanging 15 messages to release the UE context.

The registration delay is critical to 5G because authentication precedes any service. When the UDM is under attack, a legitimate UE may be denied a handover; when a UDM is under attack, the users' QoE is impacted. 5G aims to offer ultra-reliable low latency communications (URLLC) and support mission-critical services. However, as long as the CN is vulnerable to registration storm attacks, none of these services can be guaranteed. As such, the network must be able to detect and block bogus registrations at their earliest phase.

V. Blockchain-assisted 5G Authentication

In this section, we present our blockchain-assisted 5G authentication protocol. The solution applies to both 5GAKA and EAP-AKA. We first provide some cryptographic primitives before delving into the details of the solution.

A. Cryptographic Primitives

1) Blockchain and Smart Contract:: Blockchain is a decentralized digital ledger technology that allows multiple parties to record and maintain a shared database securely, transparently, and immutably. The blockchain can be categorized into public, private, and consortium-based. In the public blockchain, anyone can perform read and write operations. In contrast, the private blockchain is a permissioned ledger that only one organization can write and may allow other organizations to read the blockchain. A consortium blockchain combines public and private blockchains, where multiple organizations form a collaborative alliance that can both write and read the blockchain. Performance-wise, Ethereum, a representative of the public blockchain, can only handle 15 transactions per second, while a consortium blockchain can theoretically achieve 20000 transactions per second [27]. A smart contract is a self-executing computer program that automatically executes, controls, or documents events when specific conditions are met. They are stored on a blockchain, providing a secure and transparent way of execution which reduces the need for trusted intermediates as well as fraud losses.

2) ECDSA and ecRecover: The Elliptic Curve Digital Signature Algorithm (ECDSA) uses a private-public key pair $(sk, pk = sk * G)$ derived from an elliptic curve with order n and generator G. The signer first takes a message msg and computes its hash $z = SHA1(msg)$. The signer will then generate a random nonce k, calculate the curve point $(x_1, y_1) = k \cdot G$, derives $R = x_1 \ mod \ n$, and then computes $S = k^{-1}(z + R \cdot sk) \ mod \ n$. The signature (R, S) can be verified using pk and msg.

After receiving the signature, the verifier computes

$$\begin{aligned}(z \cdot S^{-1} \cdot G + R \cdot S^{-1} \cdot pk) &= (z + R \cdot sk) \cdot S^{-1} \cdot G \\ &= (z + R \cdot sk)(z + R \cdot sk)^{-1}(k^{-1})^{-1} \cdot G \\ &= k \cdot G\end{aligned}$$

So, checking the result with R verifies the signature. From signature (R, S) and msg, one can recover possible public key by trying point P to compute $-zR^{-1} \cdot G + S \cdot R^{-1} \cdot P$, where the x-axis of P is one of $R + n, R + 2n....$ If $P == k \cdot G$, then the result will be

$$\begin{aligned}&-zR^{-1} \cdot G + S \cdot R^{-1} \cdot k \cdot G \\ &\quad = (-z \cdot R^{-1} + (z \cdot R^{-1} + sk)) \cdot G \\ &\quad = sk \cdot G = pk\end{aligned}$$

In Ethereum, an additional signature identifier V is also included in the signature to allow one to recover pk. This is so-called the *ecRecover*. The result signature (R, S, V) will be 65 bytes long using a 256-bit long key providing a 128-bit security level.

B. Key Ideas

The ultimate goal of the mitigation solution is to block the malicious UEs' registrations from overloading the UDM. Because the UDM manages the subscribers' data and has a global view of registrations no matter where they come from, it is an ideal choice to detect and identify malicious SUPIs. Because the AMFs/SEAFs are in the CN, which has a high-security requirement, and they are relatively distributed near the edge compared to UDM, they are best fitted to block the malicious UE's requests. The UDM should inform each AMF/SEAF about the malicious SUPI. However, it is inefficient to let a single UDM inform every AMF/SEAF from different operators. Besides, AMF/SEAF are not designed to receive the SUPI before the HN authenticates the UE. Thus, we propose letting blockchain act as a trusted platform so that UDM can efficiently and securely share the information with every AMF/SEAF once it detects a malicious SUPI. Because AMF/SEAF cannot reveal the SUPI, we add a second identity to the UE, which is the blockchain address. Unlike the SUPI, which is used in different protocols such as SS7, resulting in linkability issues and jeopardizing user privacy, the blockchain address should only be used in the authentication protocol to mitigate such concerns. As SUPI and long-term key K are the only necessary confidential information needed for authentication, the long-term key K is an ideal choice to be

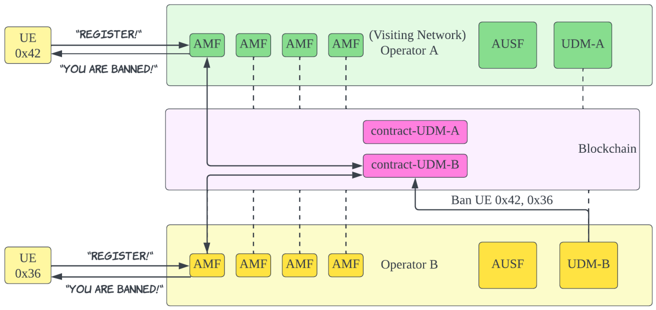

Fig. 4: System Overview

the blockchain private key that can generate the corresponding blockchain address and be updated following the Long Term Key Update Process (LTKUP). Since each UE is assigned an on-chain address, the UDM can flag the malicious UEs' addresses on the blockchain. Figure 4 shows our system model. In this example, two malicious UEs subscribed to operator B are trying to register through AMFs from operators A (visited network) and B. The UDM of operator B, UDM-B, sends the transaction to the smart contract and delegates the reject decision to AMF. The AMF checks the contract-UDM-B and sees that UDM-B has banned those UEs. As such, the AMF rejects any registration from the two UEs.

In practice, adding any extra information to the protocol can affect the performance of the 5G authentication procedure. Thus, we add an optional mode called *delegation mode* in the authentication protocol that the CN can activate when the UDM is under attack. When the delegation mode is enabled, the SEAF/AMF shall ask for UE's blockchain identity after it receives the SUCI, a slight registration delay to avoid potential high latencies. We name this extra information from the UE *Blockchain-Vector* (BV). Below are the security requirements and functional desiderata of BV to prevent a UE from bypassing the AMF if it is flagged as malicious by the UDM:

1) BV reveals the UE's blockchain address to the SEAF/AMF but not to the public.
2) BV allows SEAF/AMF to verify that the sender holds a valid blockchain account issued by its home UDM.
3) BV has a one-to-one mapping to SUCI, so one valid Blockchain-Vector cannot be replayed on any SUCI.
4) BV together with $SUCI$ cannot be replayed.
5) BV should not introduce any linkability issue.

C. The Setup

Operators with roaming agreements could set up a consortium blockchain $blockchain_{consortium}$; each operator should maintain at least one blockchain node. They should also agree on a list of public and secret key pairs (PK_{SEAF}, SK_{SEAF}) and store them in each AMF/SEAF. Each operator O_i should deploy a smart contract on $blockchain_{consortium}$ for certain UDM U_j with address $contract_j$. All these smart contracts should provide functionalities following the standard as specified in Table-I

During USIM production, operator O_i should generate the long-term key K as a random 256-bit number with a valid corresponding point on Curve25519 following the Ethereum standard. The UE holding K should be able to generate the corresponding blockchain address $address_{UE}$. In addition, operator O_i should generate another 256-bit random number $salt_{UE}$ and save it in the USIM. Then O_i updates this information in $blockchain_{consortium}$ by sending a transaction to call $\texttt{updateSalt}(address_{UE}, salt_{UE})$ in the smart contract $contract_j$. O_i should also let the UE keep a list of PK_{SEAF}.

D. Blockchain-assisted 5G Authentication Workflow

We now show the proposed Blockchain-assisted 5G Authentication protocol workflow using 5GAKA as an example. We modified the initialization phase of 5GAKA and kept the authentication phase unchanged. This way our methodology can be applied to EAP-AKA as it has the same initialization phase as 5GAKA.

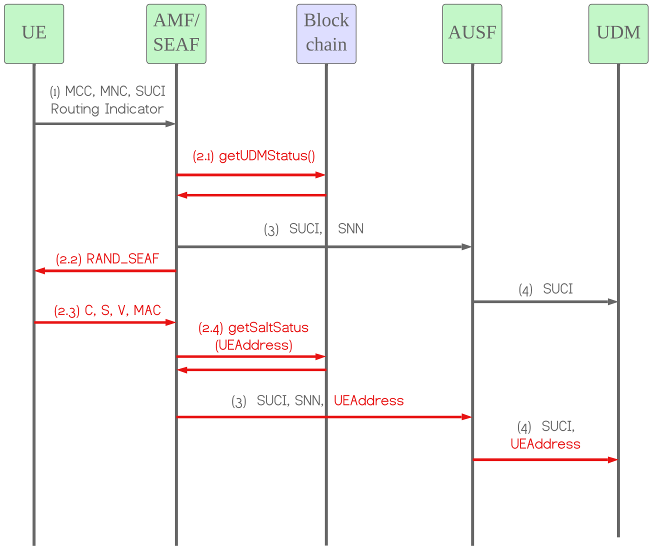

Fig. 5: Blockchain-assisted 5G Authentication: Initialization phase

Figure 5 shows our initialization phase workflow. The UE shall first send the SUCI, MCC, MNC, and a routing indicator to the SEAF. The SEAF follows Algorithm 2. First, it checks if UDM is in delegation mode. If not, the workflow follows 5GAKA. If the UDM has enabled the delegation mode, SEAF should send a random number to the UE and await the response. The UE will create the response following Algorithm 3. After receiving UE's response, the SEAF will either (i) reject the UE's registration request, or (ii) send AUSF/UDM the SUCI along with the UE's on-chain address. If UDM finds that the concealed SUPI does not match the address, then UDM bans the address and sends the transaction to the blockchain. In practice, the attacker is not motivated to sign other UE's SUCI as their account will be banned during the storm attack. In the protocol, the SEAF/AMF can periodically monitor the UDM status to avoid frequent unnecessary queries if the delegation mode is disabled. As EAP-AKA and 5GAKA

TABLE I: Smart Contract Standard

Name	Variable/Function	Input	Caller	Description
owner	variable (**addr**)	NA	NA	An address indicating the contract owner
delegate	variable (**bool**)	NA	NA	A boolean indicating if the attack is happening
blackList	variable (**addr→bool**)	NA	NA	A mapping from address to boolean. The boolean is true if the address is banned
saltList	variable (**addr→uint256**)	NA	NA	A mapping from address to integer. The number is not 0 if the address is valid
changeUDMStatus()	function	NA	owner	A **write** operation changing the boolean delegate
changeOwner()	function	**addr**	owner	A **write** operation changing the owner address
updateSalt()	function	**addr[], uint256[]**	owner	A **write** operation updating the values of addresses in saltList
banUser()	function	**addr[]**	owner	A **write** operation changing the values of addresses in blackList to True
recoverUser()	function	**addr[]**	owner	A **writ**e operation changing the values of addresses in blackList to False
getSaltStatus()	function	**addr[]**	Any	A **read** operation returns the values of addresses in saltList and blackList
getUDMStatus()	function	NA	Any	A **read** operation returns the value of delegate

Algorithm 2 AMF/SEAF Processing Logic

```
Input: $PK_{UE}$, $SK_{SEAF}$, MCC, MNC, Routing Indicator, SUCI
Locally find the UDM $contractAddress$ based on MCC, MNC, and Routing Indicator.
$underAttack = contractAddress.getUDMStatus()$
if not $underAttack$ then
    Send SUCI to AUSF; End Process
end if
Generate random number $RAND_{SEAF}$; send it to UE.
$mask = SHA256(PK_{UE} \cdot SK_{SEAF}||RNAD_{SEAF})$
Waiting to receive $BV = (C, S, V, MAC)$ from UE.
$R = mask \oplus C$
$address = ecRecover(R||S||V, SUCI||RAND_{SEAF})$
if $address ==$ 0x0 then
    Send UE ”Recover Address Failure”; End Process
end if
$salt, ban = contractAddress.getSaltSatus(address)$
if $MD5(SUCI||R||salt)! = MAC$ then
    Send UE ”Blockchain MAC Failure”; End Process
end if
if $ban$ then
    Send UE ”Registration Reject”; End Process
end if
Send SUCI, $address$ to AUSF; End Process
```

share the same initialization phase, our methodology also applies to EAP-AKA.

E. Comparison with Baseline Mitigation Procedure

The baseline methodology is to keep the authentication protocol untouched and have the UDM process all registration requests and send a registration reject message to malicious UEs after decrypting SUCI following ECIES. Indeed both the baseline mitigation procedure and the proposed Blockchain-assisted 5G authentication solution assume that the UDM is capable of detecting if a UE is compromised and being enrolled in a signaling storm attack. Compared to this baseline solution, ours has the following advantages:

1) **Eliminating the possible traffic congestion**: As the UDM is a more centralized NF, the massive amount of HTTP requests generated by the signaling storm will likely cause traffic congestion resulting in packet loss which may cause authentication failure for legitimate UEs. The massive malicious requests shall not go to the UDM in our solution and will be dealt with in a distributed fashion, so traffic congestion should not occur.
2) **Reducing the computation load of the UDM:** 3GPP indicates that the network side cannot determine the sender's identity before computing ECIES. The massive requests would exhaust the processing power of the target UDM and slow down the response to legitimate UEs [26]. In our solution, as the SUCI shall not go to UDM, this concern is no longer relevant.
3) **Reducing the concurrent session management load of the AUSF and UDM:** Managing concurrency is always challenging for network servers. The authentication protocol requires the AUSF and UDM to save temporary identifiers to keep the communication session between UE. The spike in concurrent sessions may overwhelm the servers. In our solution, this concern is mitigated.

Algorithm 3 UE Signing Process

```
Input: SUCI, $K$, $RAND_{SEAF}$, $PK_{SEAF}$
$msg = SUCI||RAND_{SEAF}$
Sign $msg$ based on Ethereum ECDSA using $K$ resulting in $(R, S, V)$.
$mask = SHA256(SK_{UE} \cdot PK_{SEAF}||RAND_{SEAF})$
$C = mask \oplus R$
$MAC = MD5(SUCI||R||salt)$
Return $BV = (C, S, V, MAC)$
```

F. Security Analysis

In this section, we evaluate whether our solution fulfills the requirements and desiderata stated earlier. It is essential to understand that the purpose of BV is not to prevent an attacker from attempting to fail someone's authentication but to prevent an attacker from legally bypassing the AMF/SEAF check.

1) **BV reveals the UE's blockchain address to the SEAF/AMF but not to the public:** As the SEAF holds the SK_{SEAF}, only SEAF can recover R. The attackers holding (C, S, V) cannot recover the blockchain address.
2) **BV allows SEAF/AMF to verify that the sender holds a valid blockchain account issued by its home UDM:** The salt is not 0 indicating the HN acknowledges this address matches one of its subscribers.

3) **BV has a one-to-one mapping to SUCI, so one valid Blockchain-Vector cannot be replayed on any SUCI:** BV is generated from $SUCI$, a previously sent BV cannot be used on another $SUCI$.
4) **BV together with $SUCI$ cannot be replayed:** BV is also generated from $RAND_{SEAF}$. An attacker holding a valid $SUCI, BV$ pair cannot bypass the check because each registration requires a different $RAND_{SEAF}$.
5) **BV should not introduce any linkability issue:** The ECDSA signature algorithm ensures that the signature will differ even with the same message. An attacker who sends UE the same $RAND_{SEAF}$ will receive a different response.
6) **The Purpose of Salt:** In our solution, we add a random number as salt saved in `saltList` other than a boolean indicating whether the UE is a valid subscriber. Although it is theoretically infeasible to compute an existing valid blockchain account, [28] has launched a blockchain private key guessing attack that discovered 732 active private keys in Ethereum. Applying the salt prevents the attacker from bypassing the blockchain checking by guessing a valid address account.

VI. Proof-of-Concept Implementation and Evaluation

We implemented a Proof-of-Concept blockchain-assisted 5G authentication procedure in an emulated environment using Kubernetes. Free5GC v3.3.0 and UERANSIM v3.2.6 were used to emulate the 5G CN and the 5G RAN respectively. The emulation of the blockchain is based on Ganache which supports the Ethereum Virtual Machine. To integrate the blockchain into Free5GC and enable smart contract functionality, we used the go-ethereum module v1.10.26 and solidity v0.8.4. Our deployment is shown in Figure 6; two sets of UEs, each containing malicious and benign UEs, connect to two AMFs through two gNBs, and the two AMFs connect to one AUSF and blockchain. Our code is publically available at https://github.com/pzeina/5g-bv-storm.git.

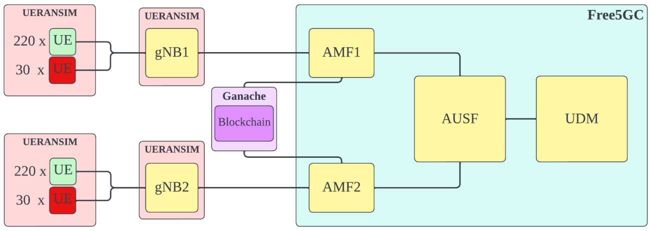

Fig. 6: Deployment Model

In our experiment, each set of UEs contains 220 benign and 30 malicious UEs. Each of the 440 benign UEs will register to the network exactly once at a randomly selected time t ranging between 1 and 250 seconds. The registration storm consists of the remaining 60 malicious UEs who will register to the network in bursts occurring once every 60 seconds. In practice, such attacks can be more continuous in time and more frequent. The smart contract was deployed in Ganache, and the AMF can query the contract storage information. We assume the UDM has previously tagged the malicious UEs and sent the transaction to the blockchain to ban these and delegate the rejection to the AMFs. Four scenarios were defined to evaluate our solution:

1) **No storm**: There is no registration storm attack.
2) **Storm with no defense**: Malicious UEs successfully register to the network and are not blocked.
3) **Storm with baseline defense**: Malicious UEs attempt to register to the network but their registration is rejected by the UDM after decryption of the SUCI.
4) **Storm with blockchain-assisted 5GAKA**: Malicious UEs are blocked by the AMFs.

Figure 7 shows the impact of the registration storm on the registration processing time under the 4 different scenarios. We measured the 440 benign UEs' registration processing time which is the difference between the authentication success timestamp and the registration start timestamp. The x-axis represents the registration start timestamps. The red vertical lines represent the start of each registration storm. The 4 sets of experiments follow the same randomly selected UEs' registration start times for fairness. The results show that the storms significantly increase the UE registration time in the absence of any defense mechanism. This is also noticeable in the presence of the baseline defense approach, as the bogus registrations still overload the UDM increasing the overall registration processing time. In contrast, our approach is more robust for it is not sensible to the storms. In fact, with our solution, we see that the registration times are very close to those in the absence of any storm. These findings are emphasized in Figure 8, which showcases the CDF of the registration time under all 4 scenarios. We see that the baseline solution mitigates the attack at a certain level but does not ideally address the problem, whereas our methodology successfully mitigates the attack at the cost of a very small delay.

VII. Conclusions

In this paper, we reviewed the 5G authentication protocol. We concluded that the de-concealment and centralized nature of the UDM can be exploited by malicious parties to initiate a registration signaling storm, overwhelming the control signal resources of the CP and leading to a denial of service for the 5G CN. To mitigate this attack, we propose a consortium blockchain-assisted solution that leverages the decentralized nature of AMF deployment to distribute the registration procedure and alleviate the load on the UDM in a trusted manner. Importantly, our approach applies to both 5GAKA and EAP-AKA protocols without affecting the performance of the authentication protocol, making it a realistic and flexible option for operators to consider. To validate the effectiveness of our solution, we conducted empirical evaluations using emulations. Our experiments provide evidence that our proposed solution outperforms the baseline mitigation approach and enhances the security and resilience of 5G networks.

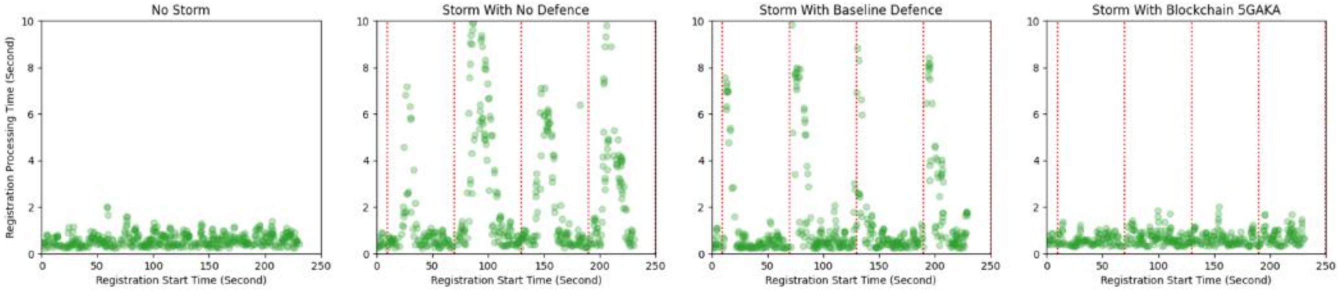

Fig. 7: Registration Processing Time Comparison for 4 Scenarios

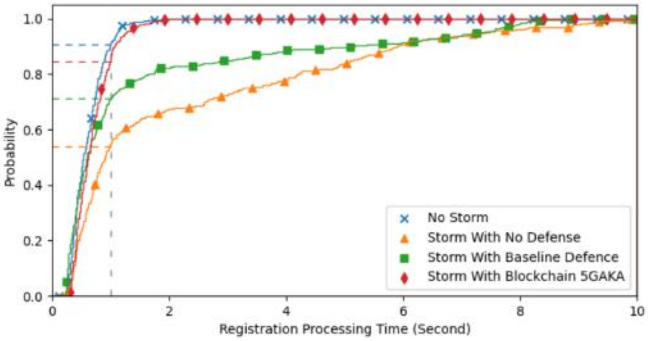

Fig. 8: CDF or registration time for the 4 Registration Storm Scenarios

Acknowledgement

This work was supported in part by funding from the Innovation for Defence Excellence and Security (IDEaS) program from the Department of National Defence (DND).

References

[1] M. Q. Khan, "Signaling storm problems in 3gpp mobile broadband networks, causes and possible solutions: A review," in *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. IEEE, 2018, pp. 183–188.

[2] P. P. Lee, T. Bu, and T. Woo, "On the detection of signaling dos attacks on 3g wireless networks," in *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*. IEEE, 2007, pp. 1289–1297.

[3] M. Pavloski, "Signalling attacks in mobile telephony," in *Security in Computer and Information Sciences: First International ISCIS Security Workshop 2018, Euro-CYBERSEC 2018, London, UK, February 26-27, 2018, Revised Selected Papers 1*. Springer, 2018, pp. 130–141.

[4] G. Gorbil, O. H. Abdelrahman, M. Pavloski, and E. Gelenbe, "Modeling and analysis of rrc-based signalling storms in 3g networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 1, pp. 113–127, 2016.

[5] T.-H. Chen, J.-W. Chang, and H.-Y. Wei, "Dynamic inter-channel resource allocation for massive m2m control signaling storm mitigation," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016, pp. 1–5.

[6] "Telenor explains mobile outage," http://www.newsinenglish.no/2011/06/16/signal-storm-caused-telenor-outages/, 2011.

[7] "Docomo to ramp up network spending following outage," https://www.mobileworldlive.com/latest-stories/docomo-to-ramp-up-network-spending-following-outage, 2011.

[8] "Docomo outage demonstrates the danger of signalling storms," https://www.verdict.co.uk/docomo-signalling-storm-outages/, 2021.

[9] G. Research, "Forecast: Internet of things, endpoints and communications, worldwide, 2020-2030, 2q21 update," https://www.gartner.com/en/documents/4004630, 2023.

[10] "State of iot—spring 2023 report by iot analytics," https://iotac.eu/state-of-iot-spring-2023-by-iot-analytics/, 2023.

[11] "8 out of top 10 cellular iot companies rely on marketsandmarkets for their growth," https://www.marketsandmarkets.com/Market-Reports/cellular-iot-market-232497754.html, 2023.

[12] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis *et al.*, "Understanding the mirai botnet," in *26th {USENIX} security symposium ({USENIX} Security 17)*, 2017, pp. 1093–1110.

[13] https://techcommunity.microsoft.com/t5/azure-for-operators-blog/what-is-the-5g-service-based-architecture-sba/ba-p/3831367.

[14] M. Pauliac, "Usim in 5g era," *Journal of ICT Standardization*, pp. 29–40, 2020.

[15] D. Strobel, "Imsi catcher," *Chair for Communication Security, Ruhr-Universität Bochum*, vol. 14, 2007.

[16] K. Ullah, I. Rashid, H. Afzal, M. M. W. Iqbal, Y. A. Bangash, and H. Abbas, "Ss7 vulnerabilities—a survey and implementation of machine learning vs rule based filtering for detection of ss7 network attacks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1337–1371, 2020.

[17] D. Basin, J. Dreier, L. Hirschi, S. Radomirovic, R. Sasse, and V. Stettler, "A formal analysis of 5g authentication," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, 2018, pp. 1383–1396.

[18] J. Cao, M. Ma, H. Li, R. Ma, Y. Sun, P. Yu, and L. Xiong, "A survey on security aspects for 3gpp 5g networks," *IEEE communications surveys & tutorials*, vol. 22, no. 1, pp. 170–195, 2019.

[19] "What is unified data management," https://www.sdxcentral.com/5g/definitions/key-elements-5g-network/what-is-unified-data-management/, 2019.

[20] Z. Gao, D. Zhang, J. Zhang, Z. Liu, H. Liu, and M. Zhao, "Bc-aka: Blockchain based asymmetric authentication and key agreement protocol for distributed 5g core network," *China Communications*, vol. 19, no. 6, pp. 66–76, 2022.

[21] M. C. Chow and M. Ma, "A secure blockchain-based authentication and key agreement scheme for 3gpp 5g networks," *Sensors*, vol. 22, no. 12, p. 4525, 2022.

[22] M. Hoffmann and P. Kryszkiewicz, "Signaling storm detection in iiot network based on the open ran architecture," *arXiv preprint arXiv:2302.08239*, 2023.

[23] S. Park, B. Cho, D. Kim, and I. You, "Machine learning based signaling ddos detection system for 5g stand alone core network," *Applied Sciences*, vol. 12, no. 23, p. 12456, 2022.

[24] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[25] A. Koutsos, "The 5g-aka authentication protocol privacy," in *2019 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2019, pp. 464–479.

[26] "3gpp: Study on authentication enhancements in 5g system," https://www.3gpp.org/ftp/Specs/archive/33_series/33.846/33846-070.zip.

[27] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, "Fastfabric: Scaling hyperledger fabric to 20 000 transactions per second," *International Journal of Network Management*, vol. 30, no. 5, p. e2099, 2020.

[28] "Ethercombing: Finding secrets in popular places," https://www.ise.io/casestudies/ethercombing/, accessed: 2019-04-23.